

Martin Krüger*, Hawo H. Höfer, David Exler, Rolf Gattung, and Markus Reischl

Noise Conditioned Bayesian Optimization Strategy Selection for Self-Driving Labs

<https://doi.org/10.1515/cdbme-2026-0172>

Abstract: Self-driving labs have become a widespread paradigm for automating research and development processes across disciplines such as biology, chemistry, and pharmaceutical research. Such systems combine robotic hardware and software to iteratively optimize experimental outcomes in a closed loop. Noise inherent in sensors and robotic actuation not only affects the speed at which the optimal result is found, but also which optimization strategies perform best. We present a simulation-based methodology to analyze self-driving lab loops and to select the optimal acquisition function and hyperparameters for Bayesian optimization by incorporating known uncertainty in realistic simulated system behavior. Our results indicate that optimizer selection informed by noise characteristics can significantly reduce the number of required iterations.

Keywords: Self-Driving Labs, Closed-Loop Optimization, Optimizer Selection, Domain-Informed Modeling.

1 Introduction

Self-driving labs (SDLs) that autonomously plan and perform experiments are becoming common across disciplines, including chemistry, biomedical, and pharmaceutical research. SDLs operate in an automated loop, aiming to optimize one or multiple properties through a series of experiments [1, 2], with applications such as the autonomous design of protein fitness for enhanced thermal stability [3]. SDLs conduct experiments using robotic hardware and measure results using sensors, while orchestration and selection of experimental parameters are performed by software [4].

As real-world experiments are often costly, optimization algorithms aim to find an optimal solution in as few steps as possible, by using knowledge from previous experiments [2, 4]. Choosing an appropriate optimizer is nontrivial, depending on factors such as the problem landscape, noise, experimental cost, and dimensionality, which vary widely across current SDL setups [2, 5, 6]. Tunable hyperparameters of opti-

mizers add further potential for optimization that is rarely used in SDL research [7, 8]. Most SDLs are based on Bayesian optimization (BO) [1, 9, 10], which constructs a surrogate model based on prior observations and selects new experimental parameters with an acquisition function. BO is also widely used in fields such as hyperparameter tuning for machine learning and drug discovery [11, 12]. Key design choices are the acquisition function and its hyperparameters, collectively referred to here as the BO strategy. Common choices include Expected Improvement (EI), Probability of Improvement (PI), and Upper Confidence Bound (UCB), which differ in how they balance exploration and exploitation. While EI and PI implicitly determine this trade-off from data, UCB allows explicit control via the exploration coefficient β [8, 11].

The BO strategy must typically be chosen prior to experiments. At this stage, researchers rely on two key knowledge sources. The first is domain knowledge, which allows limiting the range of plausible system responses. For example, ranges and sensitivities of the experimental parameters, or assumptions about smoothness, constrain the space of potential system response functions and assist in optimizer selection [13]. The second source is the hardware. Actuators (e.g. pipettes) and measurement instruments (e.g. electrochemical sensors) exhibit intrinsic inaccuracies. These inaccuracies manifest as noise, drift, and batch variability, which influence the performance of the optimization loop. As we show, the effect of both noise sources can be asymmetric and strategy dependent. A framework that models both noise sources separately is therefore necessary for reliable pre-experimental strategy selection.

SDL-specific frameworks like Olympus [6] enable benchmarking of BO and non-BO optimizers. Olympus features emulators trained on real data as well as on standardized response functions. However, such frameworks rely on a fixed set of response functions, limiting their ability to generalize findings to new SDL setups. In particular, making general statements about the influence of noise requires a large, representative set of realistic system responses to support generality. Existing frameworks are furthermore limited by the fact that noise is either learned from pre-collected data or modeled as an abstract scalar, without grounding in the physical properties of SDL components.

Gaussian Processes (GPs) define a distribution over functions via a kernel that encodes the covariance between function values based on their relative positions in input space [14].

Martin Krüger* (Corresponding author: e-mail: martin.krueger2@kit.edu), **Hawo H. Höfer, David Exler, Rolf Gattung, Markus Reischl**, Institute for Automation and Applied Informatics (IAI), Karlsruhe Institute of Technology (KIT), 76344 Eggenstein-Leopoldshafen, Germany

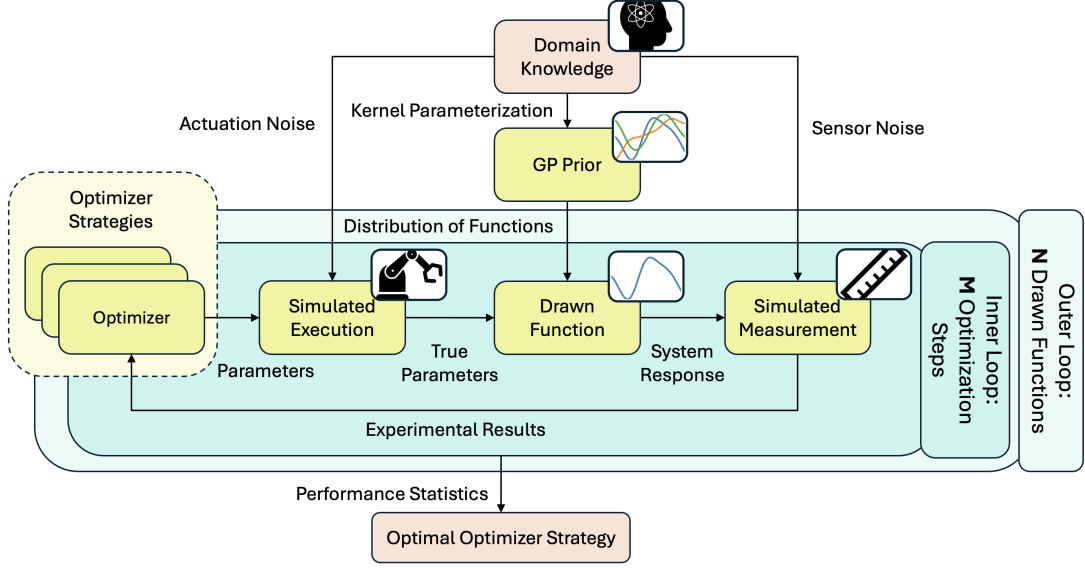


Fig. 1: Scheme of the simulation framework for optimizer selection. A GP Prior generates plausible system responses, onto which noise is applied. Candidate optimizers are evaluated in nested loops, producing Performance Statistics for optimizer strategy selection.

Functions can be sampled directly from this prior or conditioned on data to form a posterior, for example, for use as a BO surrogate model. Different kernels allow modeling structural assumptions such as periodicity, while kernel hyperparameters, in particular the length scale, control how rapidly the function varies with the input. Using one length scale per input dimension allows modeling the sensitivities independently [14–16].

Addressing the challenge of selecting a BO strategy for specific SDL setups, we propose a simulation-based methodology that utilizes known uncertainty. Unlike existing frameworks, our approach does not require pre-collected data. Instead, a GP prior is parameterized with domain knowledge, and a large number of plausible system response functions are sampled to evaluate optimizer performance across diverse plausible system behaviors. Additionally, we simulate the full SDL loop, including actuation and measurement noise. Our proposed methodology enables BO strategy selection and estimation of the experimental budget required to reach a target performance. We demonstrate the use of our framework on a generic 4-dimensional SDL and study the interaction of noise and system smoothness in this scenario.

2 Methodology

Figure 1 illustrates the proposed methodology, where a simulation abstracts a generic SDL setup consisting of a robotic actuator that sets the experimental parameters and a sensor that measures the resulting response. The user defines Actua-

tion Noise and Sensor Noise based on Domain Knowledge of the SDL hardware, and the input and output ranges of the simulated system, based on assumptions about the experiment. The space of plausible system responses is defined by Kernel Parameterization of a GP Prior, namely selecting the kernel function and specifying the range of length scales. A set of candidate Optimizer Strategies is tested in two nested loops. The Outer Loop iterates through N Drawn Functions sampled from the GP Prior, while the Inner Loop simulates one optimization run per function, searching for experimental parameters that maximize the system response within a budget of M steps. Based on the previous Experimental Results, the Optimizer predicts the Parameters $\mathbf{x}$, which represent the intended experimental settings. The Actuation Noise is applied to the Parameters, yielding the True Parameters that are used to evaluate the drawn function, simulating one conducted experiment. Sensor Noise is applied to the System Response. The Performance Statistics can be used to guide the selection of an Optimal Optimizer Strategy.

The GP prior is specified by a kernel and its hyperparameters: a fixed variance and random per-dimension length scales drawn independently and log-uniformly from a user-defined range. We sample function values on a grid of J points over the experimental parameter space via Cholesky decomposition of the kernel matrix. The resulting function is then made continuous and evaluable at arbitrary points using the GP posterior mean [14]. Since the raw function values are unbounded, the function is mapped to the experimental output range $[y_{\min}, y_{\max}]$ via

$$\hat{f}(\mathbf{x}) = y_{\min} + (y_{\max} - y_{\min}) \cdot \sigma(f(\mathbf{x})), \quad (1)$$

where $\sigma(\cdot)$ is the logistic sigmoid and $f(\mathbf{x})$ the unbounded value of the function sampled at the query point $\mathbf{x}$. Sampling N times with different length scale realizations yields a diverse set of functions covering a wide range smoothness behaviors.

Figure 2 shows two examples with two experimental parameters and different length scale configurations. The left plot uses length scales of 0.05 along the x_1 -axis and 0.3 along the x_2 -axis, resulting in a rough response, whereas the right plot (0.5 for x_1 , 1 for x_2) represents smoother system behavior.

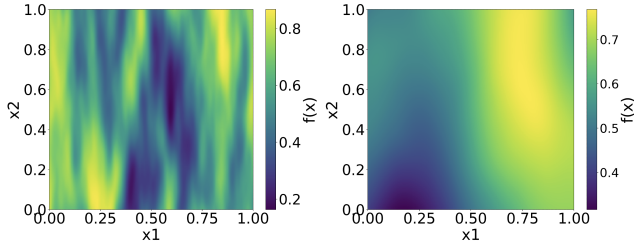


Fig. 2: Example system behavior modeled with a GP prior for different length scale configurations (left: 0.05 in x_1 , 0.3 in x_2 ; right: 0.5 in x_1 , 1 in x_2).

Noise can be modeled as Gaussian noise, where actuation noise $\epsilon_a \sim \mathcal{N}(0, \sigma_a^2)$ is applied to the predicted experimental parameters $\mathbf{x}$, and sensor noise $\epsilon_s \sim \mathcal{N}(0, \sigma_s^2)$ is applied to the system response, with standard deviations σ_a and σ_s derived from hardware characterization. Optimizer quality is assessed by the number of steps required for $\hat{f}(\mathbf{x} + \epsilon_a)$ to exceed a performance threshold, i.e., the true system response after actuation but before sensor noise is applied. Across N drawn functions, this yields a distribution of step counts per strategy.

3 Experimental setup

To demonstrate the application of our methodology on a specific SDL setup, we evaluate a generic 4-dimensional system under varying noise conditions. The simulated system has four experimental parameters and one output, all in $[0,1]$. We compare five BO strategies on $N = 500$ drawn functions: EI, PI, and UCB with $\beta \in \{0.1, 3, 12\}$. The analysis is conducted under seven noise scenarios combining $(\sigma_a, \sigma_s) \in \{(0,0), (0.01,0.01), (0.01,0.08), (0.08,0.01), (0.01,0.15), (0.15,0.01), (0.15,0.15)\}$ chosen to represent low, moderate, and high noise levels. All BO strategies are based on a GP surrogate with a Matérn 5/2 kernel. To investigate the influence of the response smoothness, we consider two length scale ranges (rough: $[0.05,0.3]$; smooth: $[0.5,1.0]$), yielding 17500 simulated runs per length scale range. We define the performance threshold as 90% of the global maximum

of each function with a maximum budget of $M = 70$ steps if the threshold is not reached. Since all evaluated BO strategies share the same GP surrogate, the Matérn 5/2 kernel shared by benchmark functions and surrogate model does not favor any acquisition function.

4 Results and Discussion

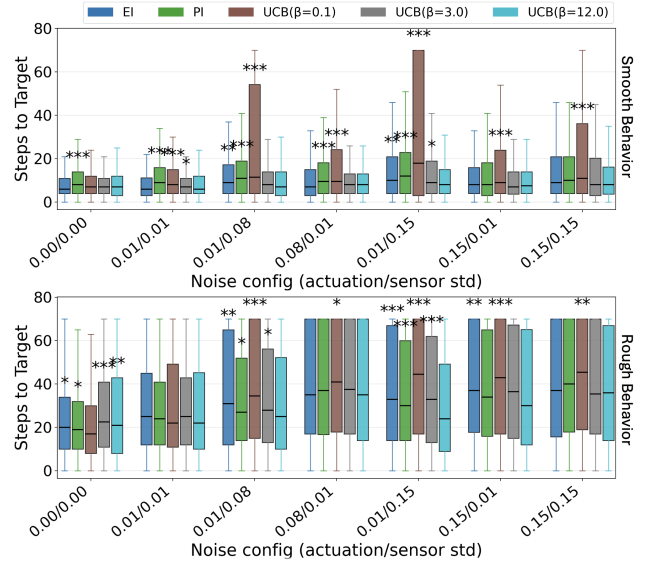


Fig. 3: Comparison of BO performance across different noise settings and function smoothness (top: higher length scales, bottom: lower length scales). The plot shows the number of steps required to reach the performance threshold.

The box-plot in Figure 3 illustrates the evaluations for two smoothness levels. The x-axis represents different actuator and sensor noise levels. Stars indicate statistically significant differences relative to the best-performing strategy in each noise configuration, based on a one-sided Mann–Whitney U test. Smooth system responses and configurations without noise lead to fast convergence, with the best strategies reaching the threshold in a median of six to eight steps, compared to around 20 steps for rough functions under the same conditions. Actuation and sensor noise both degrade optimizer performance, but their effects differ. For smooth system functions, sensor noise tends to have a greater negative impact. Especially UCB($\beta=0.1$) suffers substantially from sensor noise. The high sensor noise scenario even performs worse than a scenario with high sensor and actuation noise, in which the actuation noise can accidentally help to reach the optimum sooner. For rough system response functions, actuation noise tends to have the biggest impact on performance degradation.

The large variance across conditions reflects the stochastic nature of both drawn functions and optimization, underscoring the need for large-scale evaluation.

Overall, UCB with $\beta = 12$ performs best in most noisy scenarios in our simulation. Since high β encourages exploration, this result indicates that noisy environments benefit from strategies that are less likely to over-exploit misleading observations. In the noiseless rough environment, on the other hand, UCB with $\beta = 0.1$ outperforms all strategies, demonstrating that without noise, aggressive exploitation of the surrogate model is more effective. EI and PI consistently perform at an intermediate level across all conditions, comparable to UCB with $\beta = 3$. Researchers designing an SDL can use our methodology to simulate their systems with hardware-derived noise to compare potential BO strategies. For example, when measurement noise dominates on a rough response landscape, high-exploration UCB may be proposed.

5 Conclusion and Future Work

We developed a novel framework for simulating and analyzing SDLs without requiring pre-collected data, enabling researchers to select the optimal BO strategy and estimate the experimental budget required to reach a target performance. By simulating SDL behavior across a large number of plausible system responses using GP kernels and physically motivated noise, we showed that the optimal BO strategy depends on both noise level and system smoothness. Separating actuation and sensor noise in the model reveals asymmetric effects that cannot be captured by aggregated noise modeling and provides SDL practitioners with a more precise analysis for strategy selection. When applying this methodology to a physical SDL, known hardware noise directly parametrizes the noise in our framework, while domain knowledge about parameter sensitivity constrains the length scale range. The framework compares different strategies and can therefore recommend an optimization strategy before any real experiment is conducted.

Future work can expand the exploration of interlinks between noise, system behavior, and optimization strategies to develop and select the optimally suited algorithms. Further extensions of the methodology include hardware-motivated inaccuracies such as offsets and drift, as well as more intuitive domain knowledge modeling. Additionally, validating the framework on a real SDL represents an important next step toward practical deployment.

Author Statement

Research funding: Open access funding enabled and organized by Project DEAL. This work was funded by the program Material Systems Engineering of the Helmholtz Asso-

ciation (43.31.02). This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2154 – Project ID 390874152 (POLiS Cluster of Excellence) Conflict of interest: Authors state no conflict of interest.

References

- [1] R. Shimizu et al. Autonomous materials synthesis by machine learning and robotics. *APL Materials*, 8(11):111110, 2020.
- [2] G. Tom et al. Self-Driving Laboratories for Chemistry and Materials Science. *Chemical Reviews*, 124(16):9633–9732, 2024.
- [3] J. T. Rapp et al. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nature Chemical Engineering*, 1(1):97–107, 2024.
- [4] L. M. Roch et al. ChemOS: An Orchestration Software to Democratize Autonomous Discovery. In *Artificial Intelligence in Drug Discovery*, pages 349–388. Royal Society of Chemistry, 2020.
- [5] A. V. Tobias et al. Autonomous ‘self-driving’ laboratories: A review of technology and policy implications. *Royal Society Open Science*, 12(7):250646, 2025.
- [6] F. Häse et al. Olympus: A benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2(3):035021, 2021.
- [7] D.H. Wolpert et al. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.
- [8] B. Bischl et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, 13(2):e1484, 2023.
- [9] R. J. Hickman et al. Atlas: A brain for self-driving laboratories. *Digital Discovery*, 4(4):1006–1029, 2025.
- [10] A. E. Gongora et al. A Bayesian experimental autonomous researcher for mechanical design. *Science Advances*, 6(15):eaaz1708, 2020.
- [11] J. Wu et al. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.
- [12] H. H. Höfer et al. Bayesian Optimization Driven Cancer Drug Dose-Response Curve Discovery. *Current Directions in Biomedical Engineering*, 11(1):358–361, 2025.
- [13] K. A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Comput. Surv.*, 41(1):6:1–6:25, 2009.
- [14] C. E. Rasmussen et al. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass., 2008.
- [15] D. Khatamsaz et al. A physics informed bayesian optimization approach for material design: Application to NiTi shape memory alloys. *npj Computational Materials*, 9(1):221, 2023.
- [16] S. Pfingstl et al. On integrating prior knowledge into Gaussian processes for prognostic health monitoring. *Mechanical Systems and Signal Processing*, 171:108917, 2022.